Claims

1. A method useful in developing software having multiple versioned documents, comprising:

comparing multiple ones of the versioned documents at a plurality of different subdivision levels and indicating the changes on an output document at each subdivision level;

comparing the multiple ones of the versioned documents to a common parent document and indicating possible conflicts therebetween caused by alternative histories from the parent document;

unmerging from a later version of one of the versioned documents a set of changes previous to a further set of changes;

associating with the set of changes in the versioned documents a plurality of nonversioned documents pertaining respectively to different versions of the versioned documents;

updating one or more files by copying them from a common storage area to a private enlistment area, adding build-specific changes to the enlistment copies, making local changes to the modified enlistment copies, and thereafter removing the build-specific changes and returning the files to the common area.

2. A medium bearing representations of computer instructions and data for carrying out the method of claim 1.

3. A method useful in developing software having multiple versioned documents, comprising:

defining at least two nested types of subdivision in both of the documents;

17

for a current subdivision of the first type, comparing a current first-type subdivision of one of the documents to a current first-type subdivision of the other document, and indicating any differences therebetween;

for a current subdivision of the second type, comparing a current second-type subdivision of one of the documents to a current second-type subdivision of the other document and indicating any differences therebetween;

repeating the second comparing step for further second-type subdivisions of the documents within the current subdivision of the first type;

repeating the first comparing step for further first-type subdivisions within the documents;

producing an output document indicating the differences found in both of the comparing steps.

4. The method of claim 3 further comprising inhibiting the second comparing step when the first comparing step detects no difference between the current first-type subdivisions of the documents.

5. The method of claim 3 where the first subdivision type is a line.

6. The method of claim 3 where the second subdivision type is a character.

7. A medium bearing representations of computer instructions and data for carrying out the method of claim 3.

8. A method useful in developing software having multiple versioned documents, comprising:

defining at least three nested types of subdivision in both of the documents;

for a current subdivision of the first type, comparing a current first-type subdivision of one of the documents to a current first-type subdivision of the other document, and indicating any differences therebetween;

for a current subdivision of the second type, comparing a current second-type subdivision of one of the documents to a current second-type subdivision of the other document and indicating any differences therebetween;

for a current subdivision of the third type, comparing a current third-type subdivision of one of the documents to a current third-type subdivision of the other document and indicating any differences therebetween;

repeating the third comparing step for further third-type subdivisions of the documents within the current subdivision of the second type;

repeating the second comparing step for further second-type subdivisions of the documents within the current subdivision of the first type;

repeating the first comparing step for further first-type subdivisions within the documents;

producing an output document indicating the differences found in all of the comparing steps.

9. The method of claim 8 further comprising inhibiting the second and third comparing step when the first comparing step detects no difference between the current first-type subdivisions of the documents.

10. The method of claim 8 where the first subdivision type is a multi-line section.

11. A method useful in developing software having multiple versioned documents, comprising:

comparing two child documents of a common parent document with each other and indicating any differences as actual conflicts between the two child documents;

19

comparing both child documents with the common parent document and indicating any possible conflicts between the child documents for portions of the child documents that are the same as each other;

producing a merged output document indicating both the actual and the possible conflicts.

12. The method of claim 11 where the possible conflicts represent alternative histories of the child documents from the parent document.

13. The method of claim 11 where the possible conflicts are marked in the merged document.

14. The method of claim 11 where the actual conflicts are marked in the merged document.

15. A medium bearing representations of computer instructions and data for carrying out the method of claim 11.

16. A method useful in developing software having multiple versioned documents, comprising:

receiving one of said documents at first, second, and third version levels, where the third version level incorporates a second set of changes from the second version level, and the second version level incorporates a first set of changes from the first version level;

unmerging the first set of changes from the one document at the third version level, while preserving the second set of changes;

outputting the unmerged document

17. The method of claim 16 further comprising designating the unmerged document as a fourth version level of the one document.

18. The method of claim 16 further comprising receiving indications of the first, second, and third version levels from a user.

19. A medium bearing representations of computer instructions and data for carrying out the method of claim 16.

20. A versioned output document for developing software, the output document containing changes from one previous version of the document, but excluding changes from another previous version of the same document.

21. The document of claim 20 where the other previous version of the document is a previous version of the one document.

22. A method useful in developing software having multiple versioned documents, comprising:

     associating like versions of the versioned documents with each other in a change-set specification;

     associating additional, non-versioned documents into the same change set;

     retrieving both the versioned and nonversioned documents as a single unit.

23. The method of claim 22 where associating the versioned documents comprises listing each of them in an association file.

24. The method of claim 23 where associating the nonversioned documents comprises listing them in the same association file as the versioned documents.

25. The method of claim 23 further comprising storing the association file in a memory separately from the documents listed in the file.

26. A medium bearing representations of computer instructions and data for carrying out the method of claim 22.

27. An association file for a set of versioned documents, comprising:

a plurality of entries each designating a version of one of the versioned documents;

at least one entry designating a nonversioned document pertaining to at least one of the versioned documents.

28. The file of claim 27 where the nonversioned document includes design documentation.

29. The file of claim 27 where the nonversioned document includes bug reports.

30. The file of claim 27 where the nonversioned document includes screen shots.

31. A method useful in developing software having multiple versioned documents, comprising:

synchronizing a set of files from a common storage area to a private enlistment area;

adding a set of build-specific changes to the files in the enlistment area;

making local changes to the enlistment files;

thereafter, removing the build-specific changes from the enlistment files;

returning the enlistment files to the common area.

32. The method of claim 31 further comprising repeating the method for a plurality of separate private enlistment areas.

33. The method of claim 32 where the common area is a single area for all of the separate enlistment areas.

34. The method of claim 31 where synchronizing the common-area files includes fetching a copy of the common-area files to the private enlistment area.

35. The method of claim 34 where synchronizing the common-area files further includes merging them with a set of previous local changes.

36. The method of claim 31 where adding the build-specific changes comprises:

getting the build-specific changes from a build area;

merging the build-specific changes into the enlistment files.

37. The method of claim 36 further comprising selecting a particular build from which to get the build-specific changes.

38. A medium bearing representations of computer instructions and data for carrying out the method of claim 31.